Optimal Microgrid Sizing using Gradient-based Algorithms with Automatic Differentiation

Evelise de Godoy Antunes, Pierre Haessig, Chaoyun Wang, Roberto Chouhy Leborgne

Optimal Microgrid Sizing using Gradient-based Algorithms with Automatic Differentiation

Evelise de Godoy Antunes*†, Pierre Haessig†, Chaoyun Wang† and Roberto Chouhy Leborgne*
* Federal University of Rio Grande do Sul (UFRGS), Porto Alegre/RS, Brazil
{evelise.antunes, roberto.leborgne}@ufrgs.br
† IETR - CentraleSupélec, Cesson-Sévigné, France
pierre.haessig@centralesupelec.fr

***Abstract*—Microgrid sizing optimization is often formulated as a black-box optimization problem. This allows modeling the microgrid with a realistic temporal simulation of the energy flows between components. Such models are usually optimized with gradient-free methods, because no analytical expression for gradient is available. However, the development of new Automatic Differentiation (AD) packages allows the efficient and exact computation of the gradient of black-box models. Thus, this work proposes to solve the optimal microgrid sizing using gradient-based algorithms with AD packages. However, physical realism of the model makes the objective function discontinuous which hinders the optimization convergence. After an appropriate smoothing, the objective is still nonconvex, but convergence is achieved for more that 90 % of the starting points. This suggest that a multi-start gradient-based algorithm can improve the state-of-the-art sizing methodologies.**

***Index Terms*—Automatic differentiation, gradient-based optimization, microgrid, optimal sizing**

I. Introduction

Microgrid sizing is an optimization problem whose objective is to find the optimal values of the sizing variables, which are the capacities of the power and energy storage resources (see Fig. 1). Optimality is expressed with at least one objective function and often extra objectives or constraints which are built from performance indices related to microgrid costs, environmental impact, load serving or penetration of renewable sources. All these factors must be evaluated on the project lifecycle to include the maintenance and replacements of components. The microgrid sizing problem has been studied for some decades and several classical approaches have been consolidated into convenient software packages such as HOMER [1] or DER-CAM [2]. This means that the simplest microgrid design tasks can be considered as solved problems (i.e. optimized in seconds to minutes with HOMER). By "simple task" we mean optimizing a few components using a 1-year long hourly simulation of power balance on a single bus. Still, there is an interest in finding ever more performant sizing methods in order to tackle more complex cases. Complexity comes when optimizing more than a handful of components, or components with several parameters (e.g. the orientation of PV panels which is generally considered as fixed), but also when the sizing optimization is formulated as a multistage stochastic problem to mitigate long-term uncertainty (like fuel price over 25 years) as in Fioriti *et al.* [3]. This latter work, which is very interesting from a theoretical point of view, has a limited applicability due to the reported solving time of 70 hours. Finally, the closed-source nature of programs like HOMER prevent reusing and expanding its existing parts to adapt to new design settings. Accelerating the pace of progress in the domain calls for more open source tools.

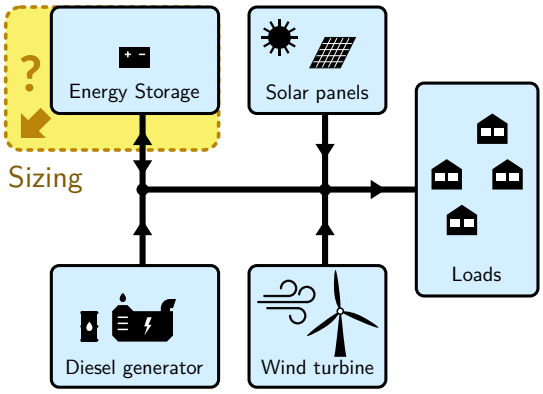

Fig. 1. Islanded microgrid architecture considered in this work. The microgrid sizing problem is about finding the optimal ratings of *all* its components.

A. Optimization approaches

There are many approaches to solve the sizing optimization problem, but they can be divided into mathematical programming (MP) and black-box optimization (BB)[1]. The MP approach consists in formulating the microgrid sizing problem algebraically, e.g., with a Mixed Integer Linear Program (MILP) model. In the BB approach, the microgrid behavior is described inside a simulator that receives as inputs the sizing variables and returns as outputs the performance indices which are used as objective and constraints functions of the optimization problem. DER-CAM [2] is a good representative of the MP approach while HOMER [1] is perhaps the most famous simulation-based, i.e. BB, sizing tool.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001, by Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq) and by the grant "Accélérer le dimensionnement des systèmes énergétiques avec la différentiation automatique" from GdR SEEDS (CNRS, France).

[1]It is a black-box optimization from the point of view of the optimization algorithm, although the cost function can be fully open source.

MP is generally formulated with modeling languages dedicated to mathematical optimization, e.g. AMPL or GAMS, or dedicated libraries, e.g. YALMIP, JuMP, Pyomo. One of the MP advantages is that the problem can be passed to reliable optimization solvers, often with fast and guaranteed convergence properties. Nevertheless, it is necessary to make several simplifications in the models of power sources and energy storage systems, e.g. linearizations, to stay in the scope of the dedicated language/library and to get the best convergence. Also, the daily operation is often optimized in an anticipative manner, i.e., disregarding the uncertainty of hourly inputs such as load or solar production, because the optimizer has access to the entire time series at once. All this can lead to a sizing that does not meet the requirements of a microgrid in practice.

The BB approach allows models which can be way more physically realistic. Indeed, it uses a temporal simulation code written with the full freedom of numerical computing languages. Also, implementing a non-anticipative operational control is easy, e.g. the Load Following strategy of HOMER [1], [4]. However, the optimization must be solved by *gradient-free* methods, for example Nelder-Mead or evolutionary algorithms [5], which could have poor convergence speed. Indeed, the objective and constraint functions are generally too complex to allow deriving their gradient by hand and using a finite difference approximation is disregarded as too slow.

However, it is possible to accelerate black-box design problems by using *gradient-based* optimization algorithms thanks to Automatic Differentiation (AD) software packages [6]. AD tools can compute the numerically exact gradient of computer codes. While AD tools have been around for some decades [7], there is a recent increase in developing high performance AD packages, because of the growing interest in the topic of "scientific machine learning" [8].

The promise of AD-assisted gradient-based microgrid sizing optimization is to obtain both fast convergence and the physically detailed models of the classical BB gradient-free approaches. Still, the non-convexity and sometimes the discontinuity of the objective function (see §III) implies that this promise should be carefully checked. To that end, our contribution is: 1) an open source microgrid simulator with a simplified HOMER-like model [9], using the high-performance Julia language [10]; 2) a gradient-based microgrid sizing tool which uses Julia's AD packages; and, 3) an analysis of the convergence speed and reliability of this method on an exemplary microgrid sizing problem.

Section II presents the proposed AD-based sizing method while section III details the microgrid model. Finally, convergence is assessed in section IV.

II. Methodology

A. Problem formulation and employed tools

The sizing optimization problem can be formulated as:

$$\min \quad f(\boldsymbol{x}) \tag{1a}$$
$$s.t. \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \tag{1b}$$
$$\boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0} \tag{1c}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of optimization variables which are n sizing parameters (ratings of PV panels, batteries...). $f(\boldsymbol{x})$ is the objective function, $\boldsymbol{h}(\boldsymbol{x})$ is the vector of equality constraints and $\boldsymbol{g}(\boldsymbol{x})$ is the vector of inequality constraints. These functions encode the performance indicators (Net Present Cost, environmental impact...) that should be optimized or constrained.

To implement AD-assisted gradient-based microgrid sizing optimization, three elements are needed:

1) a microgrid simulator
2) an Automatic Differentiation routine
3) a gradient-based optimization algorithm

Their interaction is illustrated on Fig. 2.

The microgrid simulator receives all the microgrid technical characteristics (load demand, climatic data, unit price of components...), including $\boldsymbol{x}$, the sizing to be tested. Using models presented in §III, the simulator returns the performance indicators associated with the evaluated sizing. The Automatic Differentiation routine wraps around the microgrid simulator to evaluate efficiently the gradients of the performance indicators (∇f...). Finally, the gradient-based optimization algorithm searches for the best sizing value which optimizes the performance indicators. It does so by iteratively calling the AD-wrapped simulator.

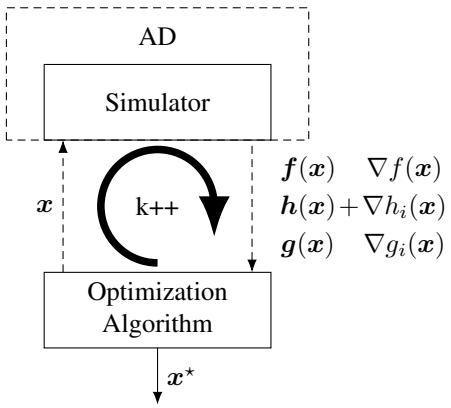

Fig. 2. Relation between optimizer and simulator.

Our proposed method is implemented using the Julia language [10] to benefit from a fast simulation speed and a rich ecosystem of AD packages. At present, we use `ForwardDiff.jl` [11] which yields fast gradients for this application (relatively small number of optimization variables to differentiate on). The gradient-based optimization algorithm is SLSQP (sequential quadratic programming) from the Julia wrapper of NLopt [12].

B. Optimization convergence assessment

Gradient-based optimization algorithms generally require the functions of (1) to be smooth (continuously differentiable) to get a reliable convergence. However, these functions correspond to microgrid performance indicators. and we show in §III that realistic HOMER-like models contain discontinuities in a few indicators.

For that reason, we introduce *relaxations* in the model to smooth the functions to enable good convergence. This smoothing creates a trade-off between the optimization convergence and model accuracy. The tuning of our proposed relaxations needs a careful evaluation of both the convergence and the accuracy of the optimization.

Our convergence & accuracy evaluation procedure is shown in Fig. 3, where $f(x)$ is the objective function for the original model and $f_{rlx}(x)$ is the objective function for the relaxed model. $\boldsymbol{x}^*$ is the optimal value of $f(x)$ (hard to obtain) while $\boldsymbol{x}^*_{rlx}$ is the optimal value found with the optimization of $f_{rlx}(x)$. The optimization is repeated for many initial points $\boldsymbol{x}^0$ with both the original and the relaxed model. Also, the relaxed optimum $\boldsymbol{x}^*_{rlx}$ is reinserted in the unrelaxed microgrid simulator to evaluate its "true" performance. We collect these data for each initial point $\boldsymbol{x}^0$ so as to see to what extent the found optimum points vary with initialization (see results in §IV).

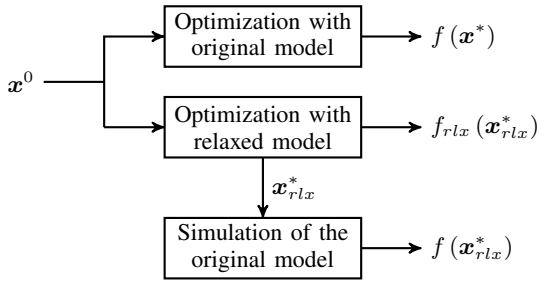

Fig. 3. Methodology for optimization convergence assessment.

Notice that this repeated procedure is only needed to evaluate the relaxation tuning. Once tuned, solving a microgrid optimization problem should only require one initial point.

III. Microgrid simulator

This section presents our new open source microgrid simulator [9], one of the three components of our sizing method (see Fig. 2). We introduce its general principle and the key performance indicators it generates. Due to space limits, we only focus on the model of one of the microgrid components, the Diesel generator, because it contains discontinuities that made us introduce a relaxation.

A. Microgrid simulator architecture

Our microgrid simulator is a simplified HOMER-style model [1] which evaluate key performance indicators of a microgrid project of the type of Fig. 1. At present, the simulator can model islanded systems with any number of nondispatchable sources (wind, PV...), one energy storage and one dispatchable generator. The simulator contains two stages:

1) An operation model that computes the energy exchanges between components using a simple rule-based energy dispatch
2) A techno-economic model that builds performance indices out of the operation data

Following HOMER, the operation model simulates one year at an hourly timestep ($\Delta t = 1\,\text{h}$, $T = 8760$ instants). The energy dispatch is a simplified Load Following [1], [4] which feeds the load P_{load} from the available sources with the following descending priority: PV, storage and then Diesel generator. As a last recourse when energy gets scarce, the load can be partially shed with power P_{shed}.

The economic model *extrapolates* the yearly operation data to compute a Net Present Cost (NPC) which sums, for each component, the initial investment, operation and replacement costs. The summation is done over the entire project lifetime and takes into account a financial discount factor. The NPC is normalized by the lifetime served energy to compute a Levelized Cost of Energy (LCOE).

The techno-economic model also computes performance indices related to the environment and the quality of service. In particular, the shedding rate (SR) is the ratio of cumulated load shedding to the cumulated desired load.

B. Dispatchable Diesel generator model

We focus on this component because it introduces two discontinuities in the project cost which call for relaxations. We implemented HOMER's generator model [1] which has a "fuel curve" (that is the fuel rate function of the generated power P_{DG}) which is affine with a discontinuity at zero. This represents the generator efficiency which tends to zero at low regime.

We detail the second discontinuity which is similar and comes from the lifetime model. Indeed, the generator is limited by a prescribed lifetime expressed as a maximum number of operation hours ℓ_{DG}. To compute the cost of replacements, the lifetime is compared against a counter of *effective* operating hours:

$$h_{DG}^{tot,proj} = \sum_{t=1}^{T} h_{DG}(t) \times \ell_{proj} \tag{2}$$

where each hour is counted whenever the generator is on:

$$h_{DG}(t) = \begin{cases} 0, & P_{DG}(t) = 0 \\ \Delta t, & P_{DG}(t) > 0 \end{cases} \tag{3}$$

Fig. 4 represents the DG operation hours model (3) which shows a discontinuity at zero. This interferes with gradient-based optimization (see IV), thus we introduce a relaxation parametrized by ε between 0 (no relaxation) and 1 (maximal relaxation):

$$h_{DG}^{rlx}(t) = \begin{cases} \dfrac{\Delta t}{\varepsilon} \dfrac{P_{DG}(t)}{P_{DG}^{rtd}}, & P_{DG}(t) \leq \varepsilon P_{DG}^{rtd} \\ \Delta t, & \varepsilon P_{DG}^{rtd} < P_{DG}(t) \leq P_{DG}^{rtd} \end{cases} \tag{4}$$

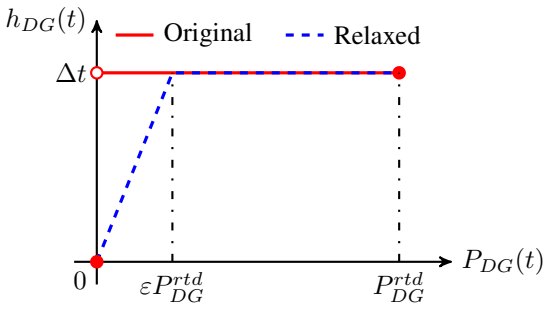

Fig. 4. Diesel generator operation hours models.

We underline that the relaxation *underestimates* the number of operation hours, and thus underestimates the replacement cost and eventually the NPC.

C. Microgrid test case

The microgrid used for the methodology validation is composed of a photovoltaic system, a battery and a Diesel generator. We use real hourly load data are from Ushant Island in 2016 [13] (2 MW peak load). We use matching solar data (hourly capacity factor time series) from the PVGIS-SARAH database [14], [15], for a south-oriented PV panel with 40° slope and 14% system loss.

The financial and technical parameters for the test project are presented in Table I. Component prices are inspired by [16], [17].

TABLE I
PROJECT AND COMPONENTS PARAMETERS.

Parameter	Symbol	Value
Project lifetime	ℓ_{proj}	25 yr
Discount factor	d	5%/yr
Battery investment price	C_{BT}^{inv}	\$ 350.00 $/kWh$
Battery operation price	C_{BT}^{OM}	\$ 10.00 $/kWh/yr$
Battery charge/discharge loss	α	5%
Battery cycling lifetime	ℓ_{BT}^{years}	15 yr
Battery calendar lifetime	N_{cycles}	3000
PV investment price	C_{PV}^{inv}	\$ 1200.00 $/kW$
PV operation price	C_{PV}^{OM}	\$ 20.00 $/kW/yr$
PV lifetime	ℓ_{PV}	25 yr
Diesel investment price	C_{DG}^{inv}	\$ 400.00 $/kW$
Diesel operation price	C_{DG}^{OM}	\$ 0.02 $/kW/h_{oper}$
Diesel lifetime	ℓ_{DG}	15,000 h_{oper}
Fuel price	C_{DG}^{fuel}	\$ 1.00 $/l$
Fuel curve slope	F_1	0.24 $L/h/kW_{output}$
Fuel curve intercept	F_0	0.00 $L/h/kW_{rated}$

IV. Convergence and Performance Assessment

To validate our sizing methodology, we need to evaluate the convergence of the optimization. Using the optimization formulation (1), the optimization variables for the microgrid sizing problem are $\boldsymbol{x} = \begin{bmatrix} P_{PV}^{rtd} & E_{BT}^{rtd} & P_{DG}^{rtd} \end{bmatrix}^T$, that is the rated power/energy of the PV, battery and Diesel generator. $NPC(\boldsymbol{x})$ is the objective function and we consider one inequality constraint, $g(\boldsymbol{x}) = SR(\boldsymbol{x}) - SR^{max}$, where SR^{max} is the maximum allowed shedding rate.

For the convergence assessment two cases were studied. First, since we introduced a relaxation parameter ε, we look at how it influences the convergence and how it biases the result to choose a reasonable value. Second, since we suspect that the maximum shedding rate SR^{max}, which is an input from the system designer, can also affect the convergence, we test a wide array of values from 0 to 5 % of shedding rate.

In both cases, the assessment is done by running the optimization with many starting points, as described in §II-B. For a most exhaustive approach, we take starting points on a regular grid in the 3D parameter space of $\boldsymbol{x}$ (min: 0, max: 10 MW(h) for P_{PV}^{rtd} and E_{BT}^{rtd}, 2 MW for P_{DG}^{rtd}, step: 500 kW(h), which makes $21 \times 21 \times 5 = 2205$ starting points).

Finally, the calculation time was also studied to analyze if this methodology could be faster than the gradient-free ones.

A. Choosing the amount of relaxation

The relaxation parameter ε is introduced to make the optimization problem continuous to ease the convergence, at the expense of biasing, i.e. underestimating, some costs, which in turn can displace the optimal sizing $\boldsymbol{x}^*$. These two aspects affect respectively the two stages of our proposed sizing method (see §II-B) and we study them separately.

1) Effect on convergence: Despite making the function continuous, the relaxation does not make it smooth or convex. Thus, the convergence of the optimization can only be assessed empirically. For choosing relaxation parameter ε, we run the gridded multistart optimization for a several amounts of relaxation between none and full (ε from 0 to 1). We conduct this analysis for one constraint level $SR^{max} = 0.01\,\%$.

For each run, we collect 2205 optimized results and we analyze the distribution of the objective and constraint functions, that are presented in Fig. 5 for $\varepsilon = 0.1$. It shows that the objective and constraint values are tightly clustered around best and maximum value respectively. However, there are some strong outliers (heavy distribution tails), that fall either in the "lower cost, unfeasible" or "higher cost, much below constraint" categories. To quantify outliers, we define tolerance thresholds: 105 % of SR^{max} for the shedding rate and 101 % of the best LCOE for the objective (tighter tolerance because the LCOE varies much less). For $\varepsilon = 0.1$, there are 0.45 % points above the objective tolerance and 1.59 % points above the constraint tolerance. Because the two cases are almost always exclusive, the total number of unacceptable solutions is the sum of both, resulting in 2.04 %.

On Fig. 6, we show that the rejection rate of optimization results generally decreases with ε. It is 100 % for $\varepsilon = 0$ (absence of convergence without relaxation) and rapidly falls to about 2 % for ε as small as 0.05. Surprisingly, the convergence decreases (up to 5 % rejection) when the relaxation becomes almost total ($\varepsilon \to 1$, which makes h_{DG} smooth, see Fig. 4). This requires further investigation.

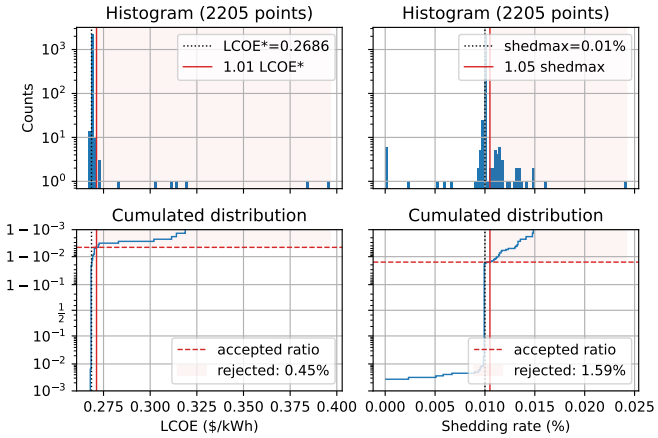

Fig. 5. Distribution of the multi-start optimization results for one test case ($\varepsilon = 0.1$, $SR^{max} = 0.01\,\%$). Distribution of the objective on the left and of the constraint on the right. Distributions are shown as log-scaled histograms on top and as cumulated distributions on the bottom, with a "logit" scale to highlight frequencies close to 0 and 1. Vertical red lines are tolerance thresholds for counting results as acceptable.

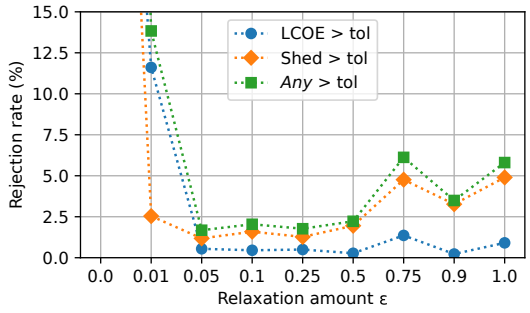

Fig. 6. Effect of the amount of relaxation ε on the rejection rate of optimization results (case $SR^{max} = 0.01\,\%$). Values for $\varepsilon = 0$ are out of bounds because all points are rejected.

2) Effect on biasing: The optimal values obtained with the relaxed objective function are used to recomputed the original model indicators, as shown in Fig. 3. The LCOE for the original and relaxed models for the tested ε are shown in Fig. 7, where the biasing effect of the relaxation can be observed. As ε increases, the difference between the original and relaxed LCOE also increases. This happens because the smoothing, due to the relaxation, in the objective function conducts the optimal point away from the original optimum.

Therefore, the higher the relaxation, the worse the result found for the original model. Using this conclusion with the results presented in the previous section, a relaxation parameter ε between 0.05 and 0.1 offers the best compromise between the effects on convergence and on biasing. For the rest of this work, we use $\varepsilon = 0.1$.

B. Robustness against the maximum shedding rate

The optimization was repeated for SR^{max} equal to 0%, 0.01%, 0.10%, 0.30%, 1.00%, 3.00% and 5.00%. The Pareto front for these SR^{max} is presented in Fig. 8. This figure

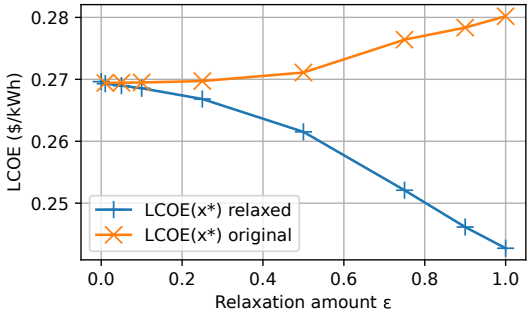

Fig. 7. Effect of the relaxation on biasing the optimal sizing (case $SR^{max} = 0.01\,\%$). The best sizing (among all multistarts) for a given amount of relaxation ε is evaluated in the unrelaxed simulator. For $\varepsilon \geq 0.25$, the relaxed optimal sizing is too far from the actual unrelaxed optimum, so that it yields a too large unrelaxed cost.

illustrates the trade-off between cost and quality of service, i.e. the LCOE decreases for higher SR^{max}, showing the coherence of the obtained results.

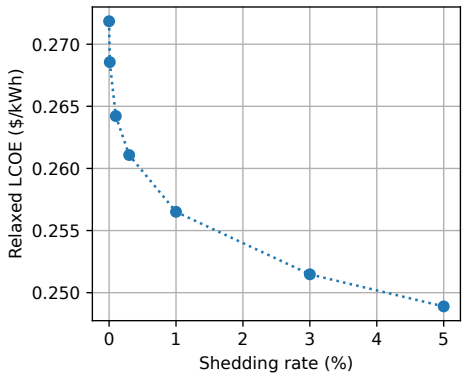

Fig. 8. Pareto front of the relaxed problem (best of all multi-starts, $\varepsilon = 0.1$).

For the studied range of SR^{max}, the rejection rates for the LCOE, shedding rate or the sum of both, do not exceed 5%. The rejections for each SR^{max} are presented in Fig. 9.

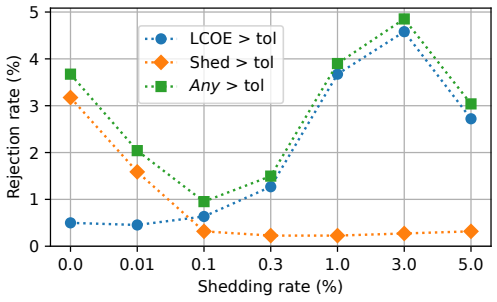

Fig. 9. Rejection rate of optimization results of the relaxed problem ($\varepsilon = 0.1$), for varying levels of shedding rate constraint SR^{max}.

Even with the methodology presenting acceptable results for this SR^{max} range, we realize that the underlying structure of the optimization problem changes since the cause of rejection varies with the shedding limit. For small levels of maximum

shedding rate, i.e. $SR^{max} < 0.1\%$, the rejection is mainly caused by the violation of the shedding rate tolerance. While for large shedding rate, the tolerance of LCOE is the primary cause of rejection.

These results suggest that using a multi-start gradient-based algorithm may be suitable for solving the optimal microgrid sizing with the proposed methodology.

C. Preliminary results of computational performance

The focus of this work is more the assessment of the accuracy (e.g. empirical convergence) of the proposed microgrid sizing method than getting the shortest possible computation time. Still, the primary motivation to use AD-based optimization is indeed the promise of a shorter computational speed, compared to gradient-free methods. So we analyze the running time of our method keeping in mind that there may be room for improvement. Also, the reference problem considered here has only 3 design parameters $\boldsymbol{x}$. The computer used for these experiments is a notebook with an Intel Core i7-9750H CPU.

The running time for the simulator alone is 15 ms on average. Computing the gradient of either the objective or the constraint is only slightly longer at 22 ms, thanks to the well-thought implementation of `ForwardDiff.jl` [11]. Now, due to the way we have interfaced our simulator to the optimization routine (NLopt SLSQP), each iteration calls separately the objective, the constraint and their gradient, which calls the simulator 4 times and makes up for 74 ms per iteration. However, there exists a better interfacing to compute all these values with one simulator call. This would bring this number down to 22 ms per iteration [18].

We measured the optimization running time with one particular representative starting point which converge in 43 iterations and this took on average 2.9 s. This is coherent with the 74 ms per iteration timing, but with the better interfacing mentioned above, this should go down to ≈ 1.0 s.

Optimizing a microgrid with 3 variables, for a 1 year long hourly simulation, in about 1 s sounds very promising, but we cannot claim this number due to the convergence difficulties studied here. A multi-start is needed and we need to check exactly how many starts are needed. Still, the results from the previous subsections suggest that only a modest number of starts is sufficient.

Also, a comparison with state-of-the-art gradient-free solvers is needed. However, only few of them truly support nonlinear constraints, e.g. NOMAD [19].

V. Conclusions

This article proposes a new approach to solve the microgrid sizing problem. We accelerate the resolution of the black-box optimization problem with gradient-based algorithms, thanks to the use of Automatic Differentiation routine.

Although the black-box method allows the use of more complex models (compared to the competing MILP based approaches), we show that some relaxations need to be implemented to smooth model discontinuities and enable the convergence of gradient-based optimization algorithms. We show that only a small amount of relaxation is needed for good convergence: a gradient-based optimization algorithm can find an optimal microgrid sizing with a reasonable tolerance level for the vast majority (95–98%) of the initial points scattered in the search space. Also, the amount of relaxation is small enough so that it does not affect too much the model accuracy.

After obtaining these promising results on a microgrid problem of small dimension (3 components to be sized), it becomes worthwhile applying the proposed gradient-accelerated sizing method more complex microgrids (more power sources and energy storage types...).

References

[1] T. Lambert, P. Gilman, and P. Lilienthal, "Micropower system modeling with HOMER," in *Integration of Alternative Sources of Energy*, F. A. Farret and M. G. Simões, Eds. John Wiley & Sons, 2005.

[2] C. Marnay, G. Venkataramanan, M. Stadler, A. S. Siddiqui, R. Firestone, and B. Chandran, "Optimal Technology Selection and Operation of Commercial-Building Microgrids," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 975–982, aug 2008.

[3] D. Fioriti, D. Poli, P. Duenas-Martinez, and I. Perez-Arriaga, "Multi-year stochastic planning of off-grid microgrids subject to significant load growth uncertainty: overcoming single-year methodologies," *Electric Power Systems Research*, vol. 194, p. 107053, 2021.

[4] C. D. Barley and C. B. Winn, "Optimal dispatch strategy in remote hybrid power systems," *Solar Energy*, vol. 58, no. 4, pp. 165–179, 1996.

[5] Y. Thiaux, J. Seigneurbieux, B. Multon, and H. Ben Ahmed, "Load profile impact on the gross energy requirement of stand-alone photovoltaic systems," *Renewable Energy*, vol. 35, no. 3, pp. 602–613, Mar. 2010.

[6] P. Enciu, F. Wurtz, L. Gerbaud, and B. Delinchant, "Automatic differentiation for electromagnetic models used in optimization," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 28, no. 5, pp. 1313–1326, sep 2009.

[7] J.-F. M. Barthelemy and L. E. Hall, "Automatic differentiation as a tool in engineering design," *Structural Optimization*, vol. 9, no. 2, pp. 76–82, apr 1995.

[8] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, "Universal Differential Equations for Scientific Machine Learning," Jan. 2020. [Online]. Available: https://arxiv.org/abs/2001.04385

[9] E. de Godoy Antunes, N. Sadou, and P. Haessig, "Microgrid.jl," https://github.com/evelisea/Microgrids.jl.

[10] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, jan 2017.

[11] J. Revels, M. Lubin, and T. Papamarkou, "Forward-Mode Automatic Differentiation in Julia," *arXiv:1607.07892 [cs.MS]*, 2016. [Online]. Available: https://arxiv.org/abs/1607.07892

[12] S. G. Johnson *et al.*, "The NLopt module for Julia," https://github.com/JuliaOpt/NLopt.jl.

[13] EDF, "Open Data de EDF sur les îles du Ponant," https://opendata-iles-ponant.edf.fr/, Accessed: 2018.

[14] European Commission Joint Research Centre, "Photovoltaic Geographical Information System (PVGIS)," https://re.jrc.ec.europa.eu/pvg_tools/.

[15] T. Huld, R. Müller, and A. Gambardella, "A new solar radiation database for estimating PV performance in europe and africa," *Solar Energy*, vol. 86, no. 6, pp. 1803–1815, jun 2012.

[16] ADEME, "Coûts des énergies renouvelables et de récupération en France. Données 2019," https://www.ademe.fr/couts-energies-renouvelables-recuperation-France, jan 2020, Accessed: 2021-05-31.

[17] Lazard, "Levelized Cost Of Energy, Levelized Cost Of Storage, and Levelized Cost Of Hydrogen," https://www.lazard.com/perspective/lcoe2020, oct 2020, Accessed: 2021-05-31.

[18] P. Haessig, "Example of NLopt+ForwardDiff to only call the "simulator" once," https://discourse.julialang.org/t/nlopt-same-complex-clalculations-in-objetive-function-and-nonlinear-constraints/41677/15.

[19] S. Le Digabel, "Algorithm 909: NOMAD: Nonlinear Optimization with the MADS Algorithm," *ACM Trans. Math. Softw.*, vol. 37, no. 4, pp. 44:1–44:15, Feb. 2011.